

**METHODS, APPARATUS AND COMPUTER PROGRAM  
PRODUCTS FOR MODELING THREE-DIMENSIONAL  
COLORED OBJECTS**

Field of the Invention

This invention relates to methods, apparatus and computer program products that model three-dimensional objects and, more particularly, to methods, apparatus and computer program products that model three-dimensional objects using texture maps.

Background of the Invention

Standard approaches to the digital representation and graphical rendering of geometric shape represent macro-scale information geometrically and micro-scale information through texture maps. These maps are arrays with hardware supported indexing and retrieval mechanisms. Texture maps that have frequently been used to add artificial detail to simple geometric models in order to make them appear more realistic include repetitive color and displacement patterns. Texture maps have also been used to encode other types of information including normal offset information. Texture maps that store normal offset information are called bump maps.

One standard approach to rendering geometric shape is disclosed in U.S. Patent No. 6,184,893 to Devic et al. entitled "Method and System for Filtering Texture Map Data for Improved Image Quality in a Graphics Computer System". In particular, the '893 patent discloses a method and system for filtering texture map data. This filtering operation is

performed to reduce flickering and sparkling when rendering a relatively small graphics primitive using a texel map of relatively larger area. Another approach to rendering geometric shape, which is described in U.S. Patent No. 5,949,424 to Cabral et al., entitled "Method, System and Computer Program Product for Bump Mapping in Tangent Space", uses a tangent space transform module. This module is used to build a tangent space transform matrix having elements comprised of normal, tangent and binormal vector components determined at a surface point on an object surface. A texture memory also stores a surface dependent or surface independent tangent space perturbed normal texture map. U.S. Patent No. 5,719,599 to Yang, entitled "Method and Apparatus for Efficient Digital Modeling and Texture Mapping", discloses methods and apparatus for mapping texture from a first raster of pixels to a second raster of pixels that may be a displayable frame buffer or other target map.

An article by M. Soucy et al., entitled "A Texture-Mapping Approach for the Compression of Colored 3-D Triangulations", The Visual Computer, Springer-Verlag, Vol. 12, pp. 503-514 (1996), addresses the problem of compressing high-resolution colored 3D triangulations into relatively compact geometric descriptions that can be displayed in real time on graphics workstations. The Soucy et al. article proposes a two step approach for the generation of compact texture-mapped models from high-resolution colored models. First, an initial triangulation is decimated using a compression technique that maintains a mapping between the vertices of the initial triangulation and a compressed 3D triangulation. Once the initial triangulation has been compressed, all vertices removed from the initial triangulation are attributed the barycentric coordinates ( $u$ ,  $v$ ,  $w$ ) of their projection onto the larger triangles in the compressed triangulation. Accordingly, each triangle in the compressed triangulation has a color triplet at each vertex, as well as a variable number of removed vertices mapped onto its planar surface. The color information that was mapped onto the compressed triangulation is then used to generate a texture map.

The texture map is tessellated so that each triangle in the compressed triangulation has its own texture domain within the texture map. Adjacent triangles on the compressed triangulation are not necessarily mapped to contiguous texture domains within the texture map in order to improve the representations of models of different topology. As a result, adjacent 3D triangles in the compressed triangulation are not necessarily adjacent once they are projected onto the texture map.

The operations described in the Soucy et al. article are limited by the fact that the texture map includes a separate texture domain for each of the large number of triangles within the compressed triangulation. This results in a large texture map and reduces rendering efficiency and speed. Moreover, because the texture map is created by mapping the vertices of the initial triangulation to the surfaces of the compressed triangulation, the images of the texture map can become distorted because of the typically poor correspondence between the initial and compressed triangulations.

An article by V. Krishnamurthy et al., entitled "Fitting Smooth Surfaces to Dense Polygon Meshes", Computer Graphics, Proceedings, pp. 313-324 (1996), discloses operations for converting dense irregular polygonal meshes (e.g., dense triangular mesh) of arbitrary topology into tensor B-spline surface patches with accompanying displacement maps. The first step in the operations consists of interactively painting patch boundaries over a rendering of the polygonal mesh. This interactive placement of patch boundaries is described as being part of the creative process and not amenable to automation. The next step involves gridded resampling of each bounded section of the triangular mesh. The resampling operation lays a grid of springs across the triangular mesh and then iterates between relaxing the grid the subdividing it. This spring grid provides a parameterization for the mesh section, which is initially unparameterized. A tensor product B-spline surface is then fit to the spring grid. A displacement map is also determined for each mesh section. This

displacement map represents the error between the tensor product B-spline surface and the spring grid.

The operations described in the Krishnamurthy et al. article are limited by the requirement of interactive painting of patch boundaries over a triangular mesh. Interactive painting involves a manual and time consuming sequence of operations to define a coarse model having quadrangular domains. Moreover, the displacement map typically provides only limited improvement in rendering because it is only computed from the difference between the final surface (B-spline surface) and the spring grid, and not between the final surface and the original triangular mesh.

Thus, notwithstanding these conventional techniques, there is a need for highly automated systems that improve shape rendering by incorporating real geometric and color detail derived directly from physical objects into enhanced texture maps. There is also a need for systems that can automatically generate light weight but highly realistic and accurate models of three-dimensional colored objects from the enhanced texture maps.

#### Summary of the Invention

Methods, apparatus and computer program products according to embodiments of the present invention can provide efficient techniques that can be used to capture actual color and shape detail from physical objects and can automatically convert this captured information into model-enhancing texture maps. These model-enhancing texture maps can then be used in conjunction with lower-complexity geometric models to generate light weight but highly realistic and accurate models of three-dimensional colored objects.

According to an embodiment of the invention, a colored model of a surface of a three-dimensional colored object may be generated from a second plurality of points that define a coarse digital representation of the surface and at least one texture map containing information derived from a first plurality of three-dimensional points that

define a fine digital representation of the surface. This derivation is achieved by mapping points within the texture map to the fine digital representation of the three-dimensional surface. Colored scan data may be used to construct the fine digital representation as a triangulated surface (i.e., triangulation) using a wrapping operation. This triangulated surface may be a two-manifold with or without nonzero boundary and the colored scan data may constitute raw point data with each datum comprising three real numbers (x-,y-, z-coordinates) providing geometric information and three integer numbers (r-,g-,b-values) providing color information. Operations are then performed to create the coarse digital representation from the fine digital representation and also preferably create a plurality a texture maps from the fine and coarse digital representations. One map may contain color information and another map may recover geometric detail lost in the simplification process associated with generating the coarse digital representation from the fine digital representation. An additional map may also be generated that corrects for differences in directions of normal vectors associated with the coarse and fine digital representations.

Preferred operations to generate a colored model of the surface of the three-dimensional colored object may also include generating a quadrangulation of the surface from the first plurality of colored points and then converting the quadrangulation into the second plurality of points. In particular, each quadrangular patch  $Q$  in the quadrangulation may be parametrized by a homeomorphism that can be constructed by decomposing the patch into a fine quadrangular grid. The grid and parametrization may then be simplified, preferably using operations that decimate the quadrangular grid through a sequence of track contractions that are prioritized by an error function. An operation may also be performed to create a coarse triangulation by decomposing each face of the simplified grid into a respective pair of triangles, with the

vertices of the resulting triangles representing the second plurality of points.

One property of the grid lines and quadrangular patch layout is that continuous lines that completely wrap the model are created.

5 Because each patch edge is connected to one and only one other patch, and each patch edge is divided into the same number of segments, the grid lines seamlessly transfer in u,v space from patch to patch. If the model is a closed manifold, each grid line will be a continuous loop. Two parallel grid lines can be thought of as a grid "track". These tracks have  
10 the same continuous property as the grid lines do. The collection of tracks defines the shape of the model just as the collection of triangles does. Each track contributes a definable amount to the general shape of the model and therefore can be prioritized in shape contributing order (based on the error function). The tracks that contribute the least amount to the  
15 shape can be systematically removed to reduce grid complexity. After each track removal, the remaining tracks are reassessed as to their shape contribution and reprioritized. This process is continued until the desired amount of complexity is reached.

Alternatively, the operations to convert the quadrangulation  
20 into the second plurality of points may include decomposing the quadrangulation into a quadrangular grid and then generating an intermediate triangulation from the quadrangular grid. The intermediate triangulation may be generated by decomposing the quadrangular grid with a constraint that vertices of the quadrangular grid and patch boundaries  
25 from the quadrangulation be preserved. The intermediate triangulation may then be decimated using a sequence of edge contractions that preserve patch boundaries from the quadrangulation.

A texture map may also be determined for the coarse digital representation of the surface by determining, for a first texel in the texture  
30 map, a respective texel coordinate that identifies a first spatial point on the coarse digital representation of the surface. The first spatial point is then

projected to a first object point on the fine digital representation of the surface. The texture map may constitute: (i) a color map by assigning a color associated with the first object point to the first texel, (ii) a displacement map by assigning a height difference between the first spatial point and the first object point to the first texel and/or (iii) a perturbed normal map by assigning a difference between a first normal at the first spatial point and a second normal at the first object point to the first texel.

If the first plurality of colored points constitute vertices of a fine triangulation and the second plurality of points constitute vertices of a coarse triangulation, then the color associated with the first object point  $y$  on the fine triangulation may be determined by a color function  $\chi(y)$ , where  $\chi(y) = \alpha\chi(a) + \beta\chi(b) + \gamma\chi(c)$ ,  $abc$  is a triangle on the fine triangulation that contains the first object point  $y$  and  $\alpha, \beta, \gamma$  are the barycentric coordinates of  $y$  defined such that  $\alpha + \beta + \gamma = 1$  and  $\alpha a + \beta b + \gamma c = y$ . Operations to determine the displacement map may be performed by assigning a height difference  $\delta(y)$  between the first spatial point  $x$  and the first object point  $y$  to the first texel, where  $y = x + \delta(y) \cdot \mathbf{n}_x$  and  $\mathbf{n}_x$  is a unit vector that extends in a direction from the first spatial point  $x$  to the first object point  $y$  and is normal to the coarse digital representation at the first spatial point  $x$ . Operations to determine the perturbed normal map may include constructing an orthonormal frame  $(N, T, B)$  defined for  $N = \mathbf{n}_x$ , where  $T$  and  $B$  are the tangent and binormal directions at the first spatial point  $x$ , respectively. A triplet  $(\lambda, \mu, \nu) \in [-1, +1]^3$  may also be determined, where  $\mathbf{n}_y = \lambda \cdot N + \mu \cdot T + \nu \cdot B$  and  $\mathbf{n}_y$  is a unit normal vector at the first object point  $y$  as expressed in the orthonormal frame. In the event the patch boundaries of the quadrangulation trace  $u$ -coordinate and  $v$ -coordinate lines, then  $T = N \times d$  and  $B = T \times N$ , where  $d$  is a direction of a  $v$ -coordinate line passing through the first spatial point on the coarse triangulation.

Additional embodiments of the invention may include operations to generate a NURBS model from the first plurality of colored

points and then convert the NURBS model into the second plurality of points. Operations to determine a texture map may also include operations to construct a texture map for the coarse digital representation using a parametrization  $\Psi$  of quadrangular patches  $Q$  on the quadrangulation. These operations to construct a texture map may also include determining for a first texel in the texture map a respective texel coordinate that, using  $\Psi^{-1}$ , identifies a first spatial point  $x$  on the coarse digital representation of the surface.

#### Brief Description of the Drawings

FIG. 1 is a flow diagram of operations performed by embodiments of the invention that can generate light weight but highly realistic and accurate models of three-dimensional colored objects.

FIG. 2 is a flow diagram of operations performed by additional embodiments of the invention.

FIG. 3 illustrates a surface piece of a fine quadrangular grid decomposed into twenty four quadrangles.

FIG. 4 illustrates the surface piece of FIG. 3, after an open whisker has been removed.

FIG. 5 illustrates a texture map containing 16 texture domains of 64 texels/domain.

FIG. 6A illustrates the mapping of a texture domain in a texture map to a corresponding patch  $Q$  and further to the fine triangulation.

FIG. 6B illustrates the preferred mapping of texture domain A in a texture domain to a corresponding patch A in a quadrangulation that is surrounded by adjacent patches B-J, in the case where texture domains B-J, which map to patches B-J, are not immediately adjacent to the texture domain A in the texture map.

FIG. 7 illustrates a local searching operation (without marking) being performed in a counter-clockwise cycle.



FIG. 8 illustrates a miss between a normal line at  $x$  and the fine triangulation  $K$ .

FIG. 9 illustrates performance of a local search operation in the projection  $K'$  of the fine triangulation  $K$ .

5 FIG. 10 illustrates orientation of a unit normal  $\mathbf{n}_y$  within an orthonormal frame  $(N, T, B)$  defined for  $\mathbf{n}_x = N$ .

FIG. 11 is a general hardware description of a computer workstation comprising software and hardware for generating highly realistic and accurate models of three-dimensional colored objects according to embodiments of the present invention.

#### Description of a Preferred Embodiment

The present invention now will be described more fully hereinafter with reference to the accompanying drawings, in which preferred embodiments of the invention are shown. This invention may, however, be embodied in many different forms and applied to other articles and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art. The operations of the present invention, as described more fully hereinbelow and in the accompanying figures, may be performed by an entirely hardware embodiment or, more preferably, an embodiment combining both software and hardware aspects and some degree of user input. Furthermore, aspects of the present invention may take the form of a computer program product on a computer-readable storage medium having computer-readable program code embodied in the medium. Any suitable computer-readable medium may be utilized including hard disks, CD-ROMs or other optical or magnetic storage devices. Like numbers refer to like elements throughout.

Various aspects of the present invention are illustrated in detail in the following figures, including flowchart illustrations. It will be understood that each of a plurality of blocks of the flowchart illustrations,

and combinations of blocks in the flowchart illustrations, can be implemented by computer program instructions. These computer program instructions may be provided to a processor or other programmable data processing apparatus to produce a machine, such that the instructions which execute on the processor or other programmable data processing apparatus create means for implementing the functions specified in the flowchart block or blocks. These computer program instructions may also be stored in a computer-readable memory that can direct a processor or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including instruction means which implement the functions specified in the flowchart block or blocks.

Accordingly, blocks of the flowchart illustrations support combinations of means for performing the specified functions, combinations of steps for performing the specified functions and program instruction means for performing the specified functions. It will also be understood that each of a plurality of blocks of the flowchart illustrations, and combinations of blocks in the flowchart illustrations, can be implemented by special purpose hardware-based computer systems which perform the specified functions or steps, or by combinations of special purpose hardware and computer instructions.

Referring now to FIG. 1, operations **100** performed by embodiments of the invention can generate light weight but highly realistic and accurate models of three-dimensional colored objects from model-enhancing texture maps and low complexity geometric models. As illustrated by Blocks **120** and **140**, operations are preferably performed to generate a fine triangulation from colored point data and generate a quadrangulation from the fine triangulation. Operations may then be performed to generate a coarse triangulation from the quadrangulation and generate at least one texture map from the fine and coarse triangulations, Blocks **160** and **180**. The coarse triangulation and at least one texture

map can then be used to generate a highly-accurate color model of a corresponding three-dimensional colored object.

According to one embodiment of the invention, a colored model of a surface of a three-dimensional colored object may be generated from a second plurality of points that define a coarse digital representation of the surface and at least one texture map containing information derived from a first plurality of colored points that define a fine digital representation of the surface. This derivation is achieved by mapping points within the texture map to the fine digital representation of the three-dimensional surface. As illustrated by the operations 100' of FIG. 2 and Blocks 110 and 120 therein, colored scan data may be used to construct the fine digital representation as a fine triangulated surface (i.e., triangulation) having colored vertices. A wrapping operation is preferably used to construct the fine triangulated surface from the colored scan data. The triangulation may be a two-manifold with or without nonzero boundary and the colored scan data may constitute raw point cloud data with each datum comprising three real numbers (x-,y-, z-coordinates) providing geometric information and three integer numbers (r-,g-,b-values) providing color information. The fine triangulation may also be generated using conventional operations that do not include wrapping or a triangulation having colored vertices may be predefined.

The colored scan data may be generated using conventional techniques for scanning physical objects. Preferred initial operations to construct the fine digital representation as a fine triangulation are more fully described in commonly assigned U.S. Application Serial No. 09/248,587, filed February 11, 1999, entitled "Method of Automatic Shape Reconstruction", the disclosure of which is hereby incorporated herein by reference. These initial operations may also include techniques to generate a Delaunay complex of colored point cloud data points. Techniques to generate Delaunay complexes are more fully described in U.S. Patent No. 5,850,229 to Edelsbrunner et al., entitled "Apparatus and

Method for Geometric Morphing", the disclosure of which is hereby incorporated herein by reference. The initial operations to construct the fine digital representation preferably use geometric techniques to infer the shape of the model from a set of colored data points that are stored in a point cloud data file. These operations may include building a Wrap<sup>TM</sup> model from the colored data point set using strict geometric rules that result in a triangulation that passes through points in the point set. Such operations are provided by commercially available software, Geomagic Studio<sup>TM</sup> 3.1, marketed by Raindrop Geomagic, Inc. of Research Triangle Park, NC, assignee of the present application.

Referring still to FIG. 2, operations may then be performed to create the coarse digital representation from the fine digital representation and also preferably to create a plurality of texture maps from the fine and coarse digital representations, as illustrated by Blocks **140**, **160** and **180**. One map may contain color information and another map may recover geometric detail lost in the simplification process associated with generating the coarse digital representation from the fine digital representation, Blocks **182** and **184**. An additional map may also be generated that corrects for differences in directions of normal vectors associated with the coarse and fine digital representations, Block **186**.

As illustrated by Block **142**, a coarse quadrangulation of the surface is preferably generated from the first plurality of colored points. Each quadrangular patch  $Q$  in the coarse quadrangulation is preferably parametrized by a homeomorphism  $q: Q \rightarrow [0,1]^2$  that may be constructed by decomposing each patch  $Q$  into a fine quadrangular grid, Block **144**. Examples of operations to generate the coarse quadrangulation from a triangulation and decompose the coarse quadrangulation into a fine quadrangular grid are more fully described in commonly assigned U.S. Application Serial No. 09/607,122, filed June 29, 2000, entitled "Methods, Apparatus and Computer Program Products for Automatically Generating Nurbs Models of Triangulated Surfaces Using Homeomorphisms", now

U.S. Patent No. \_\_\_\_\_, the disclosure of which is hereby incorporated herein by reference. Such operations are provided by commercially available software that automatically shapes the wrapped colored data points, Geomagic Studio™ 3.1, marketed by Raindrop Geomagic, Inc. of Research Triangle Park, NC. Conventional operations may also be used to generate a parametrized coarser quadrangulation with patches Q.

The fine quadrangular grid and parametrization may then be simplified, Block **162**. This simplification operation is preferably performed by decimating the fine quadrangular grid through a sequence of track contractions that are prioritized by an error function. The tracks that contribute the least amount to the shape can be systematically removed to reduce grid complexity. After each track removal, the remaining tracks are reassessed as to their shape contribution and reprioritized. Each face of the simplified grid may then be decomposed into a respective pair of triangles, with the vertices of the resulting coarse triangulated surface representing the second plurality of points, Block **164**. Because each one of the two possible diagonals across each face of the simplified grid will result in a respective pair of distinct triangles, the decomposition operation will result in one of two possible pairs of triangles for each face. As described herein, each face of the simplified grid constitutes a quadrangular patch that may be treated as the union of one of the two pairs of triangles in its corresponding triangulation. Alternatively, the operations of Block **160** to generate a coarse triangulation from the quadrangulation may include decomposing the quadrangulation into a quadrangular grid and then generating an intermediate triangulation from the quadrangular grid without explicitly decimating the quadrangular grid using track contractions. The intermediate triangulation may be generated by decomposing the quadrangular grid with a constraint that vertices of the quadrangular grid and the boundaries of the patches Q from the quadrangulation be preserved. The intermediate triangulation may then be decimated into the coarse triangulation using a sequence of edge

contractions that preserve patch boundaries from the quadrangulation. Conventional triangle-based simplification operations are described in an article by M. Garland and P.S. Heckbert, entitled "Surface Simplification Using Quadratic Error Metrics", Computer Graphics Proceedings, SIGGRAPH, pp. 209-216 (1997). These conventional operations may be used so long as they preserve the quadrangular patch structure. Patch structure can be preserved using the topology preserving constraints described in an article to T.K. Dey, H. Edelsbrunner, S. Guha and D.V. Nekhayev, entitled "Topology Preserving Edge Contraction", Publ. Inst. Math, (Beograd) (N.S.) Vol. 66, pp. 23-45 (1999), the disclosure of which is hereby incorporated herein by reference. Other parametrized representations of patches, such as NURBS and subdivision surfaces, may also be used by redefining the maps  $\Psi$  and  $\phi$  described hereinbelow. A parametrized representation of one type of subdivision surface is described in an article by A. Lee, H. Moreton and H. Hoppe, entitled "Displaced Subdivision Surface," Computer Graphics Proceedings, SIGGRAPH, pp. 85-94, (2000).

Operations are also performed to construct at least one texture map from the fine and coarse triangulations, Block **180**. These operations include determining, for a first texel in the texture map, a respective texel coordinate that identifies a first spatial point on the coarse triangulation. The first spatial point is then projected to a first object point on the fine triangulation. As illustrated by Block **182**, a color map may be determined by assigning a color associated with the first object point to the first texel. In addition, a displacement map may be constructed by assigning a height difference between the first spatial point and the first object point to the first texel, Block **184**. A perturbed normal map may be determined by assigning a difference between a first normal at the first spatial point and a second normal at the first object point to the first texel, Block **186**. Conventional operations may then be performed to store,

display, print and/or generate a color model of an object from the light weight coarse triangulation and texture map(s), Block **190**.

Referring now to FIG. 11, a general hardware description of a computer workstation is illustrated comprising, among other things, software and hardware for generating highly realistic and accurate models of three-dimensional colored objects according to embodiments of the invention. The workstation **20** preferably includes a computer **15** that may accept a point cloud data representation of a colored object via a file **19**, disk input **23** or data bus **27**. The computer **15** preferably comprises a computer-readable storage medium having computer-readable program code embodied in the medium. This computer-readable program code is readable by one or more processors within the computer **15** and tangibly embodies a program of instructions executable by the processor to perform the operations described herein and illustrated by the accompanying figures.

More detailed operations to simplify the fine quadrangular grid and parametrization, Block **162**, will now be described. These operations include simplifying the fine quadrangular grid to a coarser grid approximating the same surface. In order to efficiently construct texture maps, it is preferable to convert the coarser grid into a triangulation by decomposing each quadrangular face of the coarser grid into a respective pair of triangles. This grid decomposition operation can be performed to generate a coarse and sufficiently regular triangulation that can be decomposed into long triangle strips. These sufficiently regular triangulations are preferred because they can be rendered faster than other triangulations (e.g., irregular triangulations).

An important structural property of the fine quadrangular grid is that its dual graph can be decomposed into edge disjoint curves, which can be referred to as whiskers. If a first quadrangle of the fine quadrangular grid is entered through a first edge and exited through an opposite edge (thus entering an adjacent second quadrangle), then the

traversal path through the first quadrangle and through a sequence of second and subsequent quadrangles traces a curve. If the traversal path defined by this curve reenters the first quadrangle through the first edge, then the curve represents a closed curve (closed whisker) which may or may not cross itself. However, if the traversal path defined by the curve exits the fine quadrangular grid at a boundary edge (without reentering the fine quadrangular grid), then the curve is extended backwards by reversing the traversal direction until another boundary edge is reached, thus completing an open curve (open whisker) with two ends. FIG. 3 illustrates a surface piece of a fine quadrangular grid decomposed into twenty four (24) quadrangles, as illustrated by the solid lines. This surface piece contains one closed whisker and six open whiskers defined by dotted lines. As illustrated by FIG. 3, each of the quadrangles is traversed exactly twice, either by two cross whiskers or by a single self-crossing whisker. The crossings are in one-to-one correspondence with the quadrangles, implying that there are equally many of both. A track is the dual of a whisker, namely a sequence of quadrangles traversed by a whisker. The sequence of quadrangles may be cyclic or may start at one quadrangle and end at another quadrangle. A lateral edge of the track is one that crosses the dual whisker. The fine quadrangular grid can be simplified by the successive removal of whiskers.

Operations to simplify the fine gridded representation of the parametrized coarse quadrangulation generated at Block 142 of FIG. 2, will now be described. As will be understood by those skilled in the art, the fine quadrangular grid is a finer quadrangulation than the parametrized coarse quadrangulation and is used to capture geometric detail. The simplification operations performed on the fine quadrangular grid are preferably driven by a priority queue that orders whiskers by the error their removal causes. Removing a whisker corresponds to contracting all quadrangles of the dual track. This operation is performed by moving the two track boundaries towards each other until they meet. In other words,



all lateral edges of the track are contracted. FIG. 4 illustrates the surface piece of FIG. 3 after an open whisker has been removed. Because the removal of a whisker is equivalent to contracting the dual track, the neighboring tracks of FIG. 4 are widened. The simplification operations are preferably performed with a constraint that whisker removals not be considered in the priority queue if such removal changes the topology of the quadrangulated surface (i.e., the surfaces before and after a respective whisker has been removed are not homeomorphic to each other). Conventional operations for performing edge contractions that do not change topological type are more fully described in the aforementioned articulated to T.K Dey et al. Operations for performing edge contractions to simplify surfaces consisting of triangles in  $\mathbb{R}^3$ , which are connected together along shared edges and vertices, are also described in an article by H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stützle, entitled "Mesh Optimization", Computer Graphics, Proc. SIGGRAPH, pp. 19-26 (1993).

Because the removal of a whisker is equivalent to contracting all lateral edges of the corresponding dual track, the error associated with the removal of a whisker can be obtained by determining errors associated with contracting lateral edges of a dual track. If  $ab$  is a lateral edge, the error introduced by contracting  $ab$  to a new vertex  $c$  is measured by adding the square distances of  $c$  from a collection of planes associated with  $a$  and with  $b$ . The initial planes associated with a vertex are the ones spanned by the vertex and two contiguous neighbors in a circular order around the vertex. As described in the aforementioned article to M. Garland et al., the collection of planes are not explicitly stored but rather implicitly represented by the fundamental quadrics, which are 4-by-4 matrices from which the sums of square distances can be computed. The error of the contraction of  $ab$ , denoted as  $E(ab)$ , is defined as the minimum error achievable by varying  $c$ . If  $L$  represents a set of lateral edges of a corresponding track, then the mean, variance and mean square of the error over all edges are:

$$\text{mean} = \sum_{ab \in L} (E(ab)) / \text{card } L \quad (1)$$

$$\text{variance} = \sum_{ab \in L} (E(ab) - \text{mean})^2 / \text{card } L \quad (2)$$

$$\text{mean square} = \sum_{ab \in L} E(ab)^2 / \text{card } L \quad (3)$$

5 where the mean square is equivalent to a sum of the variance and a square of the mean. The prioritization of the tracks for removal preferably uses the mean square of the error, because this value typically gives better results than both the mean error and the maximum error defined as  $\max\{E(ab) | ab \in L\}$ . The mean square can be determined by accumulating  
10  $\sum E(ab)^2$  in a single loop over the lateral edges of the track.

A parametrization  $\Psi: Q \rightarrow [0,1]^2$  of the quadrangular patches in the quadrangulation generated at Block 140 of FIG. 2 is preferably used to construct the texture maps described with respect to Block 180. Each texture map can be viewed as associating a regular grid of points in  $[0,1]^2$  with information that is then transported to  $Q$  via the inverse map  $\Psi^{-1}$ . A  
15 two-dimensional texture array is a square grid of  $2^n$  times  $2^n$  smaller squares referred to as texels. A texel is the smallest addressable unit in the texture array. The texture array is also preferably decomposed into smaller arrays of  $2^m$  times  $2^m$  texels, for  $m < n$ , referred to as texture  
20 domains. Each such texture domain is associated with a respective quadrangular patch on the quadrangulation. As illustrated by FIG. 5, up to  $(2^{n-m})^2$  texture domains can be embedded in the texture array. In particular, FIG. 5 illustrates 16 texture domains of 64 texels/domain. This combination yields a texel array having 1024 texels. Texels are preferably  
25 addressed by their row and column within a respective texture domain, rather than in the larger embedded texture array. Given the parametrization  $\Psi: Q \rightarrow [0,1]^2$ , a center of a texel is preferably mapped to a corresponding patch  $Q$  using  $\Psi^{-1}$ . Points besides or in addition to the center can also be used in the mapping. In particular, each texel in row  $k$

and column  $l$  can be identified by the point:

$$t_k^l = ((k+1/2)/2^m, (l+1/2)/2^m) \in [0,1]^2 \quad (4)$$

where  $0 \leq k, l \leq 2^m-1$ . The corresponding point of  $Q$  is  $\Psi^{-1}(t_k^l)$ , as illustrated by FIG. 6A, where each texel center is mapped to its preimage on a respective patch  $Q$ .

Each vertex of a triangle within a patch  $Q$  has a  $u,v$  texture coordinate assigned to it that indexes into the texture map. The  $u,v$  texture coordinate range in the quadrangulation is from 0.0 to 1.0. The coordinate 0.0, 0.0 is the lower left-hand corner of the texture map with 1.0, 1.0 being the upper right-hand corner. Since each triangle in the coarse triangulation is generated from a grid line track intersection, and the grid lines represent the  $u,v$  patch parametrization (which also goes from 0.0, 0.0 to 1.0, 1.0), the triangle vertices have definable  $u,v$  coordinates with the respective patch.

The parametrization  $\Psi: Q \rightarrow [0,1]^2$  can be constructed using the representation of each quadrangular patch  $Q$  as a grid of  $I \cdot J$  quadrangular faces arranged in  $I$  rows and  $J$  columns. There are  $I + 1$  rows and  $J + 1$  columns of vertices, and for each  $0 \leq i \leq I$  and  $0 \leq j \leq J$ ,  $R_i$  and  $C^j$  are defined to be the paths of edges connecting the vertices of row  $i$  and the vertices of column  $j$ . The vertex at the intersection of the two paths is  $q_i^j = R_i \cap C^j$ . The coordinates of  $q_i^j$  are by definition the fractions of the initial paths,

$$u(q_i^j) = |R_{0i}^j| / |R_i| \quad (5)$$

$$v(q_i^j) = |C_{0i}^j| / |C^j| \quad (6)$$

where  $|R_{0i}^j|$  is the length of the path between  $q_0^j$  and  $q_i^j$ ,  $|R_i| = |R_{0i}^j|$ ,  $|C_{0i}^j|$  is the length of the path between  $q_0^j$  and  $q_i^j$ , and  $|C^j| = |C_{0i}^j|$ . Since each

face of the grid is preferably decomposed into two triangles, as illustrated by FIG. 6A, the definition of coordinates can be extended to all points of  $Q$  using linear interpolation. Accordingly, the definition  $\Psi(x) = (u(x), v(x))$  applies for all points  $x \in Q$ .

5                   The grid representing a patch  $Q$  is constructed using relaxation methods that enable a layout with gradual variation of length and angle. This gradual variation of length and angle is one reason why the map  $\Psi$  is, in most cases, a homeomorphism and can therefore be used as a parametrization. It is possible, however, that  $\Psi$  is not a homeomorphism. 10 This happens if and only if the vertices of a positively oriented triangle  $pqr$  in the representation of  $Q$  map to points  $\Psi(p)$ ,  $\Psi(q)$ ,  $\Psi(r)$  with negative orientation in  $[0,1]^2$ . When this happens, the preimage of every point  $x \in [0,1]^2$  still exists but may be ambiguous. This is because  $\Psi$  is at least surjective, as noted by the fact that  $\Psi$  is continuous by construction and that it maps the boundary of  $Q$  surjectively onto the boundary of  $[0,1]^2$ . 15

Operations to determine the preimage  $x = \Psi^{-1}(t_k^l)$  of each texel center include a local search starting in the triangle that contains  $\Psi^{-1}(t_{k-1}^{l-1})$ , and if  $l=0$ , then the triangle that contains  $\Psi^{-1}(t_{k-1}^l)$  is the starting point of the local search operation. However, if  $k=l=0$ , then the starting point may be the upper left corner triangle of  $Q$ . Often,  $x$  will lie in the same triangle, in which case its barycentric coordinates can be computed by solving a 3-by-3 linear system of equations. Otherwise, an operation is performed to step from triangle to triangle until a triangle containing  $x$  is found. Assuming  $\Psi$  is a homeomorphism, this triangle  $pqr$  containing  $x$  is 20 unique and characterized by the fact that  $t_k^l = \Psi(x)$  lies on the positive sides of the directed lines passing through  $\Psi(p)$ ,  $\Psi(q)$ , through  $\Psi(q)$ ,  $\Psi(r)$ , and through  $\Psi(r)$ ,  $\Psi(p)$ , in this order. Any other triangle has an edge  $pq$  such that  $\Psi(x)$  is on the negative side of the directed line passing through  $\Psi(p)$ ,  $\Psi(q)$ . The search can thus continue in the triangle on the other side of  $pq$ . 25 30

However, even if  $\Psi$  is homeomorphic, it is possible that these search operations may run in circles, as shown in FIG. 7. In particular, FIG. 7 shows sixteen shaded triangles that form a counter-clockwise cycle and may be followed by the local search operations, if the search operations are performed without marking. Additional complications may be caused by the possibility of a non-homeomorphic map  $\Psi$ . To avoid infinite loops, it is preferable that triangles be marked when they are visited and that marked triangles not be visited twice during the same search. The resulting strategy is a depth-first search that typically hones in on the correct triangle rather quickly. In a worst case scenario, this strategy exhausts all triangles in the representation of  $Q$  and is therefore guaranteed to find a triangle that contains a preimage of  $t_k^l$ .

It is preferred that the information stored in the texture maps (color, displacement and perturbed normal) be obtained from the most accurate source available, which is the original (colored) fine triangulation  $K$  of a surface of a colored object. As illustrated by Block **120** of FIG. 2, this triangulation is preferably generated by wrapping colored data points. This information is accessed by mapping each spatial point  $x = \Psi^{-1}(t_k^l)$  on each patch  $Q$  to a corresponding object point  $y = \phi(x)$  on  $K$ . For a point  $x \in Q$ ,  $N_x$  can be treated as the line normal to  $Q$  that passes through  $x$ . The object point  $y = \phi(x)$  is defined as the point where  $N_x$  intersects  $K$ . The mapping operation  $\phi$  is referred to as the normal projection of the coarse triangulation, Block **160**, onto the fine triangulation, Block **120**. Since both the coarse and fine triangulations approximate the same surface,  $x$  is generally well-defined, but in exceptional cases it may happen that  $N_x$  intersects  $K$  in more than one point or not at all. If the former case,  $\phi(x)$  can be defined as the point  $y \in N_x \cap K$  closest to  $x$ . An example of the latter case is shown in FIG. 8, where the normal line at  $x$  is illustrated as missing the triangulation  $K$  represented by the solid curve. In this latter case,  $\phi(x)$  can be defined as the object point  $y$  on  $K$  that minimizes the

distance to  $x$ . Since the exceptional cases are typically rare, the influences on the texture maps is typically unnoticeable.

The operations for determining  $\phi$  are similar to the operations for determining  $\Psi^{-1}$ . Let  $x$  be a spatial point on a patch  $Q$  and assume that a search for a nearby point  $z \in Q$  has just been performed. The triangle  $abc \in K$  is used that contains  $\phi(z)$  as the starting point for the search. Let  $K'$  be the projection of  $K$  along the direction of  $N_x$ , as illustrated in one dimension in FIG. 9. Finding  $\phi(x)$  by searching locally from triangle to triangle starting at  $\phi(z)$  is generally fast and straightforward but may be complicated because (i) there may be configurations of triangles in  $K'$  that are as shown in FIG. 7, (ii) the triangulation  $K$  may fold over itself in the projection along  $N_x$ , or (iii) the line  $N_x$  may intersect  $K$  in more than one point or it may miss  $K$ . A marking operation, as described above with respect to the operations for determining  $\Psi^{-1}$ , is sufficient to deal with complications (i) and (ii). If  $N_x$  intersects  $K$  in more than one point, then the search operations will find one of these multiple points but not the one necessarily closest to  $x$ . However, since the search proceeds locally, the search will typically find a point sufficiently close to  $x$ . Finally, in the case where  $\phi(x)$  is unacceptably far from  $x$  or  $\phi(x)$  cannot be found, the following relationship will be satisfied:  $N_x \cap K = \emptyset$ . In this case, operations will be performed that find  $y = \phi(x)$  by exhaustively searching  $K$ . These operations are permitted to be relatively slow since they are infrequently performed.

Texture maps are preferably constructed using the maps  $\Psi$  and  $\phi$ . In the cases of constructing a color map, a displacement map and a perturbed normal map, the texture information of a texel with center  $t$  is computed from the corresponding point  $y$  on  $K$ , where:

$$y = \phi(\Psi^{-1}(t)) \quad (7)$$

The color map stores color information. The standard representation of color typically uses 24 bits, 8 bits each to specify one of  $2^8 = 256$  possible red, green and blue values. Because each vertex  $a \in K$  has a color value  $\chi(a)$  by construction, the color of the object point  $y \in K$  can be determined by linear interpolation as:

$$\chi(y) = \alpha\chi(a) + \beta\chi(b) + \gamma\chi(c) \quad (8)$$

where  $abc$  is a triangle on the fine triangulation  $K$  that contains the first object point  $y$  and  $\alpha, \beta, \gamma$  are the barycentric coordinates of  $y$  defined such that  $\alpha + \beta + \gamma = 1$  and  $\alpha a + \beta b + \gamma c = y$ . Since  $\alpha, \beta, \gamma \geq 0$ , a valid color  $\chi(y)$  is rounded to the nearest integer and stored in the texel with center  $t$ . Other operations that can determine the color of the object point from the color of the vertices of the triangle  $abc$  may also be used.

In cases where the fine triangulation  $K$  is significantly finer than the texel array, the quality of the color map can be improved by performing an operation to oversample and average color. In particular, rather than using only the texel center, the center point and four corner points of the texel are mapped to  $K$ . The color value to be stored in the texel can then be obtained by determining and averaging the color of the five corresponding object points on  $K$ . The operations for determining the color map may also account for rendering artifacts that can be observed when several texels are mapped to the same pixel of an image device displaying the color object, Block 190. The display mechanism automatically averages the color values of all texels in an appropriate neighborhood of the center texel, which leads to artifacts when texture domains that are not adjacent in the texel array map to adjacent patches of the model surface. To eliminate or at least reduce these display artifacts, the first and last rows and columns of every texture domain (i.e., peripheral texels) can be used to duplicate color information from texture domains that correspond to other patches, as illustrated by FIG. 6B. These other

patches extend adjacent a patch to which interior texels within a corresponding texture domain are mapped. This partial duplication insures that color averaging operations performed by the display device are performed over an appropriate collection of color values. FIG. 6B also illustrates how peripheral texels, such as corner texels, may receive color information based on an average of a plurality of adjacent patches.

The displacement map stores local offset information. The standard representation uses 24 bits but typically repeats the same 8 bits three times, so only 256 different values can be encoded. Each value approximates a multiple of the unit outward normal. To construct the displacement map,  $\mathbf{n}_x$  is defined as the unit outward normal at the spatial point  $\mathbf{x} = \Psi^{-1}(\mathbf{t})$ . The object point is determined by  $\mathbf{y} = \phi(\mathbf{x})$  and the displacement  $\delta(\mathbf{y})$  is defined as the signed height difference between the spatial point  $\mathbf{x}$  and the object point  $\mathbf{y}$  and is determined by solving:

$$\mathbf{y} = \mathbf{x} + \delta(\mathbf{y}) \cdot \mathbf{n}_x \quad (9)$$

Using these relationships, the displacement values for all texel centers (or other corners, etc.) can be determined. An affine transformation can also be constructed that maps a minimum displacement value to the value 0 and a maximum displacement value to the value 255. The displacement  $\delta(\mathbf{y})$  is first transformed and then rounded down to the nearest integer in  $[0,255]$ , with the maximum also rounded down to 255. The integer value is the value stored in the texel with center  $\mathbf{t}$ .

The normal map stores the difference between the outward unit normals at the spatial points  $\mathbf{x} = \Psi^{-1}(\mathbf{t})$  and their normal projections  $\mathbf{y} = \phi(\mathbf{x})$ . The difference may also be expressed in three times 8 bits. In order to determine the difference between the outward normals and their normal projections, an orthonormal frame is preferably constructed that includes the normal vector  $\mathbf{n}_x$ . In this orthonormal frame  $(\mathbf{N}, \mathbf{T}, \mathbf{B})$  defined for  $\mathbf{n}_x = \mathbf{N}$ , the unit normal vector  $\mathbf{n}_y$  is expressed as well, as illustrated by FIG. 10. It



is preferable that the tangent direction  $T$  and binormal direction  $B$  be chosen to extend at least roughly along the  $u$ -coordinate and  $v$ -coordinate lines passing through  $x \in Q$ . More precisely, the following deterministic operations are preferably performed to obtain a predictable orthonormal frame where:

$$\begin{aligned} N &= \mathbf{n}_x \\ d &= v\text{-direction at } x \\ T &= N \times d \\ T &= T / \|T\| \\ B &= T \times N \end{aligned}$$

By construction,  $N$ ,  $T$  and  $B$  are all unit length. A triplet  $(\lambda, \mu, u) \in [-1, +1]^3$  may also be determined, where:

$$\mathbf{n}_y = \lambda \cdot N + \mu \cdot T + u \cdot B \quad (10)$$

and  $\mathbf{n}_y$  is a unit normal vector at the first object point  $y$  as expressed in the orthonormal frame. As described above with respect to the operations for constructing the displacement map, an affine transformation and rounding can be used to map  $\lambda$ ,  $\mu$ , and  $u$  to three 8-bit integers. This triplet is the value stored in the texel with center  $t$ . Rendering systems can make use of this normal offset information to make more accurate and realistic lighting calculations. The perturbed normal map can make low triangle count models (e.g., coarse triangulation) have the accurate lighting effects usually seen on much higher density models.

In the drawings and specification, there have been disclosed typical preferred embodiments of the invention and, although specific terms are employed, they are used in a generic and descriptive sense only and not for purposes of limitation, the scope of the invention being set forth in the following claims.